

IN THE CLAIMS:

Please cancel Claims 2, 4, 5 and 23 without prejudice or disclaimer of subject matter, and amend Claims 1, 3, 6, 8 to 10, 12, 16 to 18, 20 to 22, 24, 28, 29 to 31 and 33 as shown below. The claims, as pending in the subject application, now read as follows:

1. (Currently amended) A computer implemented method for generating a device driver for an output [[a]] device by [[in]] an information processing apparatus ~~that executes an application~~, wherein said output device is connected to said information processing apparatus, said method comprising the steps of:

loading an application from a read-only memory, said application including a device model independent device driver;

determining a model of said output [[a]] device to which said application is ~~intending~~ intended to issue output commands ~~interface;~~

determining whether a device model dependent configuration ~~first part of~~ data in a memory device matches said model of said output device;

upon determining that said device model dependent configuration ~~first part of~~ data in said memory device matches said model of said output device, reading ~~loading~~ said device model dependent configuration ~~first part of~~ data from said memory device; and

generating said device driver for said output device by configuring said device model independent device driver with said device model dependent configuration data ~~using said first part of data and a second part of data stored in a memory, wherein said memory stores said application and said second part of data.~~

2. (Canceled)

3. (Currently amended) A method according to claim 1 [[2]], wherein said device model dependent configuration data is color conversion data to convert RGB color-space to a native color-space of said device.

4. and 5. (Canceled)

6. (Currently amended) A method according to claim 1, wherein said application in said read-only memory is an unchangeable application.

7. (Original) A method according to claim 1, wherein said information processing apparatus is a game console.

8. (Currently amended) A method according to claim 1 [[6]], wherein said ~~unchangeable~~ application is a game application executed on a game console.

9. (Currently amended) A method according to claim 1, further comprising the step of copying said device model dependent configuration ~~first part of~~ data into a memory card connected to said information processing apparatus.

10. (Currently amended) A method according to claim 1, wherein said output device is a printer, and said memory device is a disc associated with said ~~for a~~ printer.

11. (Previously presented) A method according to claim 1, wherein said memory device is a server over a network.

12. (Currently amended) A method according to claim 1, wherein said output device is a printer, and said memory device a memory in said [[a]] printer.

13. (Canceled)

14. (Previously presented) A method according to claim 1, wherein said model of said device is determined through reading an identification string from said device.

15. (Canceled)

16. (Currently amended) A method according to claim 1, wherein said memory device is a memory card connected to said information processing apparatus.

17. (Currently amended) A method according to claim 1, wherein, upon determining that said device model dependent configuration ~~first part of~~ data in said memory device does not match said model of said output device, said determining step further determines whether further configuration ~~a third part of~~ data in a further memory device matches said model of said output device, and upon determining that said further configuration ~~third part of~~ data in said further memory device matches said model of said output device, loading said further configuration ~~third part of~~ data from said further memory device as said device model dependent configuration ~~first part of~~ data.

18. (Currently amended) A method according to claim 1, wherein said read-only memory device is an optical storage ~~[[a]]~~ disc for a game.

19. (Original) A method according to claim 1, wherein said device is connected to said information processing apparatus by means of a USB connection.

20. (Currently amended) A method according to claim 1, wherein said output device is a printer, and said loading configuration ~~obtaining~~ step is executed in the information processing apparatus when said application is controlled to perform a printing operation is ~~required by a user~~.

21. (Currently amended) An information processing apparatus for executing an application and for generating a device driver for an output ~~[[a]]~~ device connected to said information processing apparatus, said information processing apparatus comprising:

a read-only memory for storing an application, said application including a device model independent device driver;

a memory device for storing device model dependent configuration data; and

a processor for~~[[:]]~~ determining a model of said output ~~[[a]]~~ device to which said application intends to issue output commands ~~is intended to interface~~, determining whether said device model dependent configuration data ~~a first part of data in said memory device~~ matches said model of said output device, upon determining that said device model dependent configuration ~~first part of data in said memory device~~ matches said model of said output device, ~~reading~~ ~~loading~~ said device model dependent configuration ~~a first part of data~~ from said memory device and generating said device driver for said device by configuring said device model independent device driver with said device model dependent configuration ~~using said first part of data and a second part of data stored in a memory, wherein said memory stores said application and said second part of data.~~

22. (Currently amended) A computer program to be executed in an information processing apparatus for executing an application and for generating a device driver for an output ~~[[a]]~~ device connected to said information processing apparatus, said computer program being stored on a read-only memory and comprising:

a device model independent device driver;

code for determining a model of an output ~~said~~ device to which said application intends to issue output commands ~~is intended to interface~~;

code for determining whether device model dependent configuration ~~a first part of data in a memory device~~ matches said model of said output device;

upon determining that said device model dependent configuration ~~first part of data~~
in said memory device matches said model, code for reading ~~loading~~ said device model
dependent configuration ~~first part of data~~ from said memory device; and

code for generating said device driver for said output device by configuring said
device model independent device driver with said device model dependent configuration ~~using~~
~~said first part of data and a second part of data stored in a memory, wherein said memory stores~~
~~said application and said second part of data.~~

23. (Canceled)

24. (Currently amended) A method of providing forward compatibility of device
driver code of an unchangeable application with a plurality of device models, wherein said
application is stored in a read-only memory and not linked to other executable code, said method
comprising the steps of:

including device model independent device driver code in said application;

determining a model of an output ~~[[a]]~~ device which said application is desired to
issue commands to ~~interface with~~;

reading model dependent configuration data for said model of said output device;

and

generating a device driver for said model of said output device by configuring said
device model independent device driver code with said model dependent configuration data.

25. (Original) A method according to claim 24, wherein said model of said device is determined through reading an identification string from said device.

26. (Original) A method according to claim 24 or 25, wherein said unchangeable application is a game application executed on a game console.

27. (Original) A method according to claim 26, wherein said model dependent configuration data is read from a memory card of said game console.

28. (Currently amended) An information processing apparatus for providing forward compatibility of device driver code of an unchangeable application with a plurality of device models, wherein said application is stored in a read-only memory and not linked to other executable code and said device driver code is device model independent, said apparatus comprising:

storage means for storing model dependent configuration data for a plurality of devices;

means for determining a model of an output [[a]] device which said application is desired to issue commands to ~~interface with~~;

data reading means for reading model dependent configuration data associated with [[of]] said model of said output device from said storage means; and

means for generating a device driver for said model of said output device by configuring said device model independent device driver code with said model dependent configuration data.

29. (Currently amended) Apparatus according to claim 28, wherein said output device is connected to said apparatus by means of a USB connection.

30. (Currently amended) Apparatus according to claim 28 or 29, wherein said model of said output device is determined through reading an identification string from said output device.

31. (Currently amended) Apparatus according to any one of claims 28 to 30 and 29, wherein said unchangeable application is a game application and said apparatus is a game console.

32. (Original) Apparatus according to claim 31, wherein said storage means is a memory card of said game console.

33. (Currently amended) A computer program product, carried on a read-only storage medium, for providing forward compatibility with a plurality of device models, wherein said computer program product is ~~unchangeable and~~ not linked to other executable code, said computer program product comprising:

an application;

~~model independent device driver code;~~

code for determining a model of an output ~~[[a]]~~ device which said application is desired to issue commands to interface with;

code for reading model dependent configuration data associated with [[for]] said model of said output device from a memory device; and

code for generating a device driver for said model of said output device by configuring ~~said~~ device model independent device driver code with said model dependent configuration data.